College of Engineering and Technology, MODY University, Rajasthan

RESEARCH PLAN
ON
PERFORMANCE ENHANCEMENT OF SLOT CONFIGURATIONS FOR MAPREDUCE ON BIG DATA

SCHOLAR NAME : Dinesh Mishra

ENROLLMENT NUMBER : 150442

SUPERVISOR : Prof. Anil Kumar
HOD CSE Dept.

DATE OF SUBMISSION : 16/12/2017

Introduction

Big Data: Definition

Big information is a term that alludes to informational collections or mixes of informational indexes whose size (volume), many-sided quality (inconstancy), and rate of development (speed) make them hard to be caught, overseen, prepared or broke down by ordinary innovations and apparatuses, for example, social databases and desktop measurements or representation bundles, inside the time important to make them helpful. While the size used to decide if a specific informational index is viewed as large information is not immovably characterized and keeps on changing after some time, most investigators and professionals as of now allude to informational collections from 30-50 terabytes(10 12 or 1000 gigabytes for each terabyte) to different petabytes (1015 or 1000 terabytes for each petabyte) as large information. Figure No. 1 gives Layered Architecture of Big Data System. It can be deteriorated into three layers, including Infrastructure Layer, Computing Layer, and Application Layer start to finish.

Problem With Big Data Processing

Heterogeneity And Incompleteness

Volume of information: Volume alludes to measure of information. Volume of information put away in big business storehouses have developed from megabytes and gigabytes to petabytes.

Assortment of information: Different sorts of information and wellsprings of information. Information assortment detonated from organized and heritage information put away in big business archives to unstructured, semi organized, sound, video, XML and so on.

Speed of information: Velocity alludes to the speed of information preparing. For time-delicate procedures, for example, getting misrepresentation, enormous information must be utilized as it streams into your venture keeping in mind the end goal to amplify its esteem.

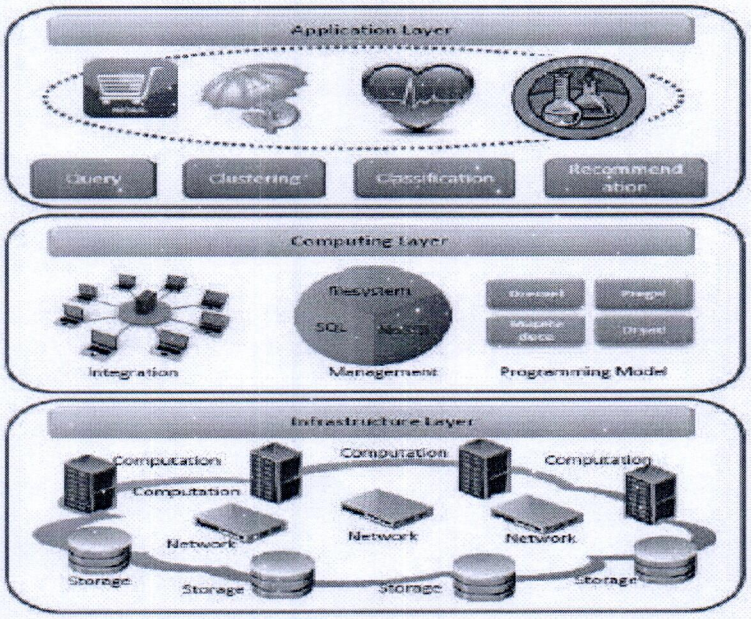


Figure 1: Layered Architecture of Big Data System

In this module , we are going to perform two processes. Slot allocation Slot pre - scheduling process. In this slot allocation process , we are going allocate the slot based on dynamic Hadoop slot allocation optimization mechanism. In the slot pre scheduling process we are going to improve the data locality. Slot Pre - Scheduling technique that can improve the data locality while having no negative impact on the fairness of Map - Reduce jobs. Some idle slots which cannot be allocated due to the load balancing constraint during runtime, we can pre - allocate those slots of the node to jobs to maximize the data locality.

Scale

Of course, the first thing anyone thinks of Big Data is its Size.Managing large and rapidly increasing volumes of data has been challenging issue for many decades.

Timeliness

The flip side of size is speed. The large the data set to be processed , the longer it will take analyze.

Privacy

The privacy of data is another huge concern, and one that increases in the context of Big Data.

Literature Survey

Year	Author	Contribution
2012	Shiori KURAZUMI *, Tomoaki TSUMURA*, Shoichi SAITO* and Hiroshi MATSUO	Dynamic processing slots scheduling for I/O and improvement of CPU performance
2012	Shanjiang Tang, Bu-Sung Lee, Bingsheng He	Dynamic Hadoop Fair Scheduler (DHFS) performance on cluster and pool level
2013	S.Vikram Phaneendra & E.Madhusudhan Reddy	Information protection
2013	Kiran kumara Reddi & Dnvsl Indira	Different Technique to Transfer Big Data
2013	Yin Li, Chuang Lin, Fengyuan Ren	The utilization of the cluster is the main concern for the system itself.
2013	Jimmy Lin	Using Hadoop" Hammer & Nails" algorithms.
2014	Umasri.M.L, Shyamalagowri.D ,Suresh Kumar.S	Mining Big Data:- Current status and forecast to the future

2014	Shanjiang Tang, Bu-Sung Lee, and Bingsheng He	Speculative Execution Performance Balancing to balance the performance tradeoff between a single job and a batch of jobs
2015	Yong Zhao, Ioan Raicu, Shiyong Lu, Cui Lin, Yanzhe Zhang, Wenhong Tian, Ruini Xue	Implementation of the framework by integrating the Swift scientific workflow management system with the Open Nebula and Eucalyptus cloud platforms
2015	Swathi Prabhu, Anisha P Rodrigues, Guru Prasad M S & Nagesh H R	Tuning methods to enhance the performance of MapReduce jobs
2015	Xiaolong Yu, Wei Li	The effect of workload (and number of Mapper slots) on the system performance (i.e., utilization) is exposed
2016	Orcun Yildiz, Shadi Ibrahim , Gabriel Antoniu	Preemption technique for pausing and resuming both map and reduce tasks
2016	Shanjiang Tang, Bu-Sung Lee, and Bingsheng He	Algorithms to minimize the makespan and the total completion time for an offline MapReduce workload
2016	Zhihong Liu, Qi Zhang, Reaz Ahmed, Raouf Boutaba, Yaping Liu, and Zhenghu Gong	Incur high performance overhead due to the partition size prediction and repartitioning
2017	Zhihui Lu , Nini Wang, Jie Wua, Meikang Qiu	IoTDeM, big data-oriented model for predicting MapReduce performances
2017	WenTai Wu , WeiWei Lin , Ching-Hsien Hsu b, LiGang He	Energy-aware data management focuses on data distribution on HDFS
2017	Mohd Usama, Mengchen Liu, Min Chen	Issues and Challenges regarding job scheduling in Hadoop
2017	Yi Yao, Jiayin Wang, Bo Sheng, Chiu C. Tan, and Ningfang Mi	How to minimize the completion length (i.e., makespan) of a set of MapReduce jobs.

Other Components of HADOOP

The Table 1, Comparison among Components of Hadoop, gives points of interest of various Hadoop Components which have been utilized now days. HBase, Hive, Mongo DB, Redis, Cassandra and Drizzle are the distinctive segments. Correlation among these segments is done on the premise of Concurrency, Durability, Replication Method, Database Model and Consistency Concepts utilized as a part of the segments.

Table 1: Comparison among Components of Hadoop

Name	HBase	Hive	MongoDB	Redis	Cassandra	Drizzle
Description	Wide-column store based on Apache Hadoop and on concepts of Big Table	Data Warehouse Software for Querying and Managing Large Distributed Datasets, built on Hadoop	One of the most popular Document Stores	In-memory Database with configurable options performance vs. persistency	Wide-column store based on ideas of BigTable and DynamoDB	MySQL fork with a pluggable micro-kernel and with an emphasis of performance over compatibility
Implementation language	Java	Java	C++	C	Java	C++
Database Model	Wide Column Store	Relational DBMS	Document Store	Key – Value Store	Wide Column Store	Relational DBMS
Consistency Concepts	Immediate Consistency	Eventual Consistency	Eventual Consistency, Immediate Consistency	-	Eventual Consistency, Immediate Consistency	-
Concurrency	Yes	Yes	Yes	Yes	Yes	Yes
Durability	Yes	Yes	Yes	Yes	Yes	Yes
Replication Method	Selected Replication factor	Selected Replication factor	Master – Slave Replication	Master – Slave Replication	Selected Replication factor	Master – Master Replication, Master – Slave Replication

Research Gap

- In all the works discussed above and the comparison in different components of Hadoop, the works on map reduce and slot configuration different algorithm only.
- The mostly works on mining of data different algorithm different parameters.
- A MapReduce job Ji computation consists of two phases, a map phase M and reduce phase R. Each phase consists of a number of tasks. We write jJMi j and jJRi j for the number oftasks in Ji's map phase and reduce phase, respectively. Let tMi;j and tRi;j denote the execution time of Ji's jth map task and jth reduce task, respectively.

- We consider a MapReduce workload with a set of independent jobs J ¼ {J1; J2; . . . Jn}, for some n. These jobs can be executed in any order. The workload is executed on a MapReduce cluster under FIFO scheduling, consisting of a set of (map and reduce) slots, denoted as S. Let SM and SR denote the set of map slots and reduce slots configured by MapReduce administrator (i.e., S ¼ SM [SR), so that the number of map slots and reduce slots are jSMj and jSRj, correspondingly.

Research Methodology

- Our alternately class of algorithms focuses on the job ordering optimization for a MapReduce workload under a if and only if map/reduce one armed bandit configuration.
- In study, our second class of algorithms considers the game plan that we can travail optimization for map/reduce one armed bandit configuration for a MapReduce workload.
- Points of interest of Proposed System:
  - We propose a bi-criteria heuristic calculation to enhance makespan and add up to fruition time at the same time.
  - Propose space setup calculations for makespan and add up to fruition time. We additionally demonstrate that there is a relative element for them, which is essential and can be utilized to address the time productivity issue of proposed list calculations for a vast size of aggregate openings

MODULES

- Job Ordering Optimization Module
- Slot Configuration Optimization Module

Module Description:

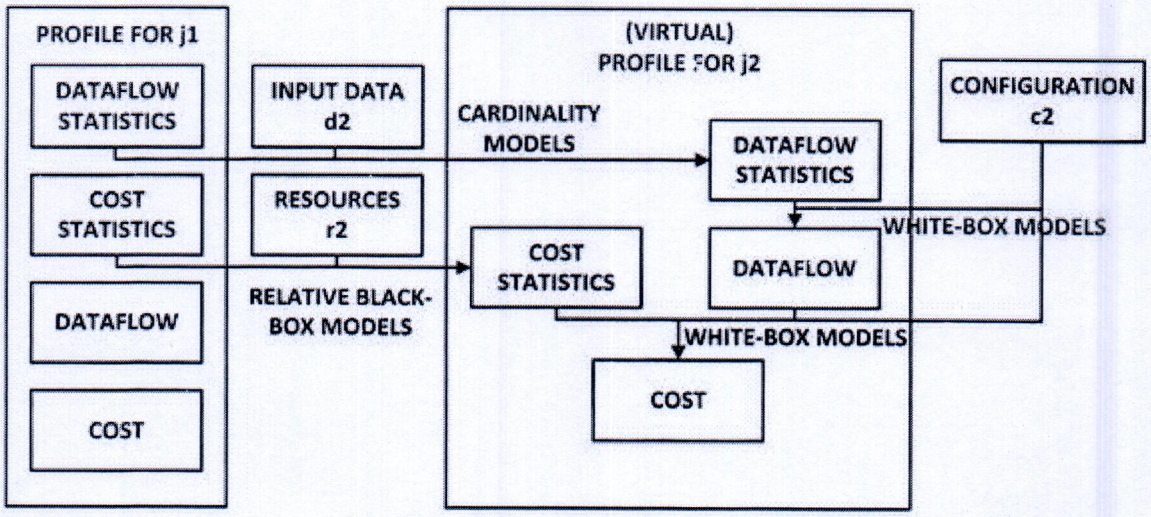
Occupation Ordering Optimization:

This module we can fundamentally enhance the aggregate finishing time of MapReduce. For Map Reducer workloads we utilize the employment requesting improvement calculations MK_JR and MK_TCT_JR.

Opening Configuration Optimization:

The opening setup can significantly affect execution for MapReduce workloads. We propose a few count calculations for delineate/space setup advancement as to the makespan and aggregate consummation time of a MapReduce workload.

- Coding language : Java/Python



Conclusion

MapReduce is a popular simulate computing pattern for large-scale front page new processing in clusters and front page new centers. A MapReduce workload consistently contains a reside of jobs, each of which consists of multiple map tasks followed by multiple minimize tasks. Due to that how to book tasks bounce only contest in reference book slots and minimize tasks can only barnstorm in cut slots, and the commander execution constraints that manual tasks are entire before abbreviate tasks, march to a diverse drummer enrollment execution orders and map/reduce one armed bandit configurations for a MapReduce workload have significantly different performance and program utilization. This complimentary proposes two classes of algorithms to cut back the makespan and the total cessation predate for an offline MapReduce workload.

Our alternately class of algorithms focuses on the job ordering optimization for a MapReduce workload under a if and only if map/reduce one armed bandit configuration. In study, our second class of algorithms considers the game plan that we can travail optimization for map/reduce one armed bandit configuration for a MapReduce workload.

REFERENCES

1. S.Vikram Phaneendra & E.Madhusudhan Reddy "Big Data- solutions for RDBMS problems- A survey" In 12th IEEE/IFIP Network Operations &

Management Symposium (NOMS 2010) (Osaka, Japan, Apr 19{23 2013).

2. Kiran kumara Reddi & Dnvsl Indira "Different Technique to Transfer Big Data : survey" IEEE Transactions on 52(8) (Aug.2013) 2348 { 2355}

3. Jimmy Lin "MapReduce Is Good Enough?" The control project. IEEE Computer 32 (2013).

4. Umasri.M.L, Shyamalagowri.D ,Suresh Kumar.S "Mining Big Data:-
Current status and forecast to the future" Volume 4, Issue 1, January 2014 ISSN: 2277 128X

5. Albert Bifet "Mining Big Data In Real Time" Informatica 37 (2013) 15–20 DEC 2012

6. Bernice Purcell "The emergence of "big data" technology and analytics" Journal of Technology Research 2013.

7. Sameer Agarwal†, Barzan MozafariX, Aurojit Panda†, Henry Milner†, Samuel MaddenX, Ion Stoica "BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data" Copyright © 2013ì ACM 978-1-4503-1994 2/13/04

8. Yingyi Bu _ Bill Howe _ Magdalena Balazinska _ Michael D. Ernst "The HaLoop Approach to Large-Scale Iterative Data Analysis" VLDB 2010 paper "HaLoop: Efficient Iterative Data Processing on Large Clusters.

9. Shadi Ibrahim⋆ _ Hai Jin _ Lu Lu "Handling Partitioning Skew in MapReduce using LEEN" ACM 51 (2008) 107–113

10. Kenn Slagter · Ching-Hsien Hsu "An improved partitioning mechanism for optimizing massive data analysis using MapReduce" Published online: 11 April 2013 © Springer Science+Business Media New York 2013.

11. Ahmed Eldawy, Mohamed F. Mokbel "A Demonstration of Spatial Hadoop:An Efficient MapReduce Framework for Spatial Data" Proceedings of the VLDB Endowment, Vol. 6, No. 12 Copyright 2013 VLDB Endowment 21508097/13/10.

12. Jeffrey Dean and Sanjay Ghemawat "MapReduce: Simplified Data Processing on Large Clusters" OSDI 2010

13. Niketan Pansare1, Vinayak Borkar2, Chris Jermaine1, Tyson Condie

"Online Aggregation for Large MapReduce Jobs" August 29September 3, 2011, Seattle, WA Copyright 2011 VLDB Endowment, ACM

14. Tyson Condie, Neil Conway, Peter Alvarc, Joseph M. Hellerstein "Online Aggregation and Continuous Query support in MapReduce" SIGMOD"10, June 6–11, 2010, Indianapolis, Indiana, USA. Copyright 2010 ACM 978-1-4503-0032-2/10/06.

15. Jonathan Paul Olmsted "Scaling at Scale: Ideal Point Estimation with „Big-Data" Princeton Institute for Computational Science and Engineering 2014.

16. Jonathan Stuart Ward and Adam Barker "Undefined By Data: A Survey of Big Data Definitions" Stamford, CT: Gartner, 2012.

17. Balaji Palanisamy, Member, IEEE, Aameek Singh, Member, IEEE Ling Liu, Senior Member, IEEE" Cost-effective Resource Provisioning for MapReduce in a Cloud"gartner report 2010, 25

18. Mrigank Mridul, Akashdeep Khajuria, Snehasish Dutta, Kumar N " Analysis of Bidgata using Apache Hadoop and Map Reduce" Volume 4, Issue 5, May 2014" 27

19. Kyong-Ha Lee Hyunsik Choi "Parallel Data Processing with MapReduce: A Survey" SIGMOD Record, December 2011 (Vol. 40, No. 4)

20. Chen He Ying Lu David Swanson "Matchmaking: A New MapReduce Scheduling" in 10th IEEE International Conference on Computer and Information Technology (CIT"10), pp. 2736–2743, 2010

21. Yong Zhao, Ioan Raicu, Shiyong Lu, Cui Lin, Yanzhe Zhang, Wenhong Tian, Ruini Xue, "A Service Framework for Scientific Workflow Management in the Cloud" IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 8, NO. 6, NOVEMBER/DECEMBER 2015

22. Xiaolong Yu, Wei Li, "Performance Modelling and Analysis of MapReduce/Hadoop Workloads" 978-1-4673-6762-2/15/$31.00 c 2015 IEEE

23 Shanjiang Tang, Bu-Sung Lee, and Bingsheng He," Dynamic Job Ordering and Slot Configurations for MapReduce Workloads" IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 9, NO. 1, JANUARY/FEBRUARY 2016

24 Zhihong Liu, Qi Zhang, Reaz Ahmed, Raouf Boutaba, Yaping Liu, and Zhenghu Gong," Dynamic Resource Allocation for MapReduce with Partitioning Skew" IEEE TRANSACTIONS ON COMPUTERS, VOL. 65, NO. 11, NOVEMBER 2016

25. Zhihui Lu , Nini Wang, Jie Wua, Meikang Qiu "An IoT Big Data-oriented MapReduce performance prediction extended model in multiple edge clouds "J. Parallel Distrib. Comput. (2017), https://doi.org/10.1016/j.jpdc.2017.11.001.

26. WenTai Wu , WeiWei Lin , Ching-Hsien Hsu , LiGang He "Energy-Efficient Hadoop for Big Data Analytics and Computing: A Systematic Review and Research Insights" Future Generation Computer Systems (2017)

27. Mohd Usama, Mengchen Liu, Min Chen ," Job schedulers for Big data processing in Hadoop environment: testing real-life schedulers using benchmark programs" 2352-8648/© 2017 Chongqing University of Posts and Telecommunications. Production and hosting by Elsevier

28. Yi Yao, Jiayin Wang, Bo Sheng, Chiu C. Tan, and Ningfang Mi, Self-Adjusting Slot Configurations for Homogeneous and Heterogeneous Hadoop Clusters" IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 5, NO. 2, APRIL-JUNE 2017